

SNS Remote Operations Experience + Thoughts on Using Java

John Galambos
SNS, ORNL

Remote Operations Workshop

Sept. 17-20, 2002

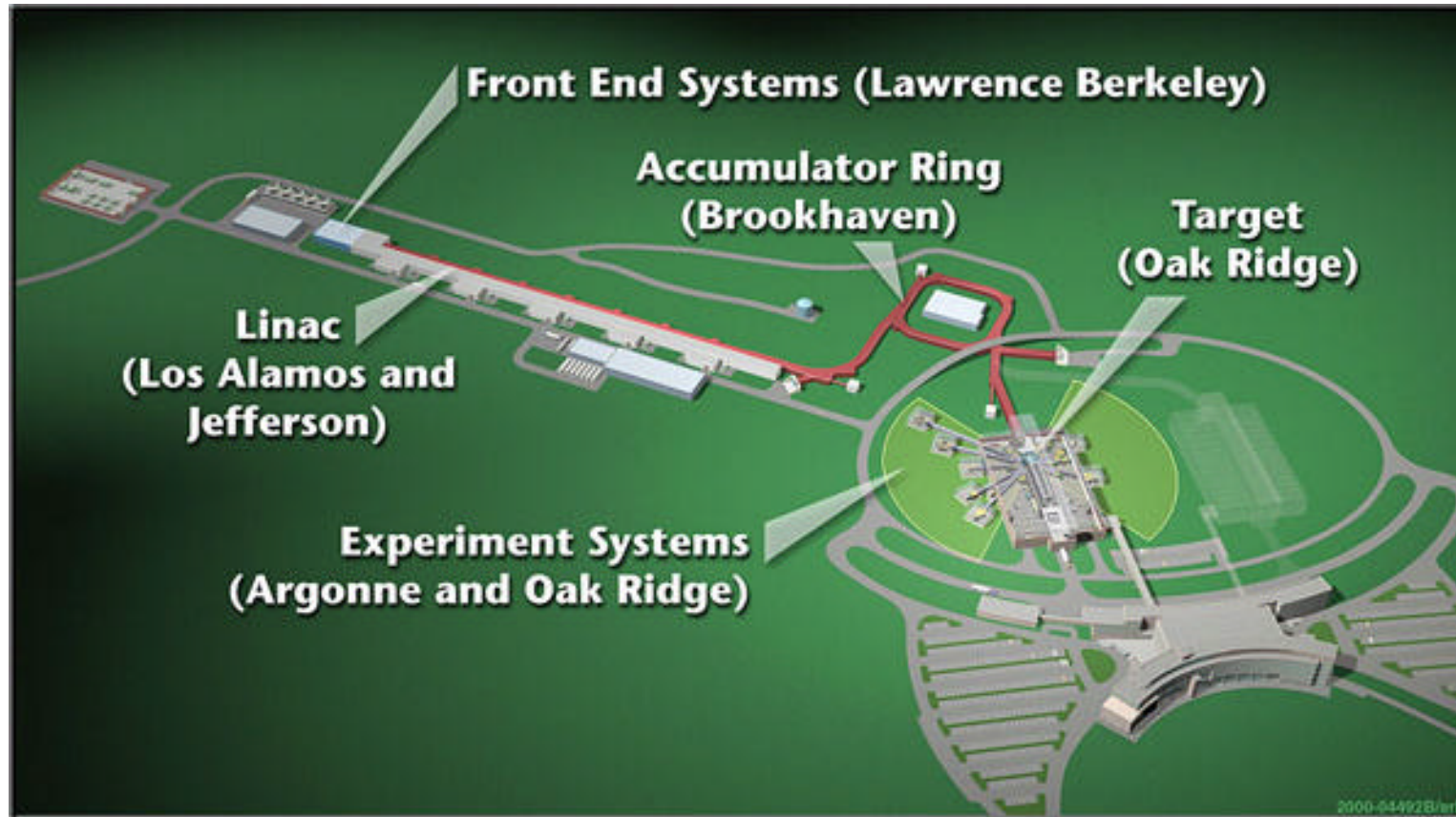
Outline



- SNS remote testing of high level applications
 - During initial commissioning of SNS MEBT at LBNL
- XAL java based high level application programming infrastructure
 - Implications for remote operations

Acknowledgements: C. Allen, S. Aleksandrov, P. Chu, N. Pattengale, T. Pelaia, A. Shishlo, E. Williams, C. Lionberg, N. Malitsky

SNS is a Multi-lab Collaboration



- Accelerator components provided by LBNL, LANL, Jlab, and BNL

Remote Testing of Applications



Front End
at LBNL



Test control room
at ORNL



- Remote testing of high level applications from ORNL on the Front End System at LBNL

SNS Medium Energy Beam Transport (MEBT) Section



- Transport section from RFQ to the DTL
 - Provides chopping and beam matching functionality
 - 3.65 m long
- Beamline equipment
 - 14 quads
 - 6 vertical + 6 horizontal correctors
 - 4 RF rebunches
 - 6 BPMs, 2 current monitors, 4 profile monitors, 1 emittance measurement device
- Beam Parameters
 - Energy = 2.5 MeV
 - Nominal current = 36 mA (commissioning current ~ 20 mA)
 - Nominally 1 msec pulse length (commissioning pulse length ~ 100 μ sec)
 - Nominally 60 Hz (commissioning rate ~ 1 Hz)

SNS Remote Testing of High Level Physics Applications



- The SNS MEBT was commissioned at LBNL April-May 2002.
- The initial schedule included time for testing high level applications, but:
 - Required components were not functional when our time approached,
 - Many others were at LBNL
 - Advised not to come
- Through steady “pestering” we were able to secure 3 slots for testing our applications.
 - “Bootleg” operation
 - Total of ~ 5 hrs beam-time
 - First 2 slots had no “set” privilege
- Tested model comparison, orbit correction + general purpose diagnostic app.

SNS Remote Applications Testing Setup

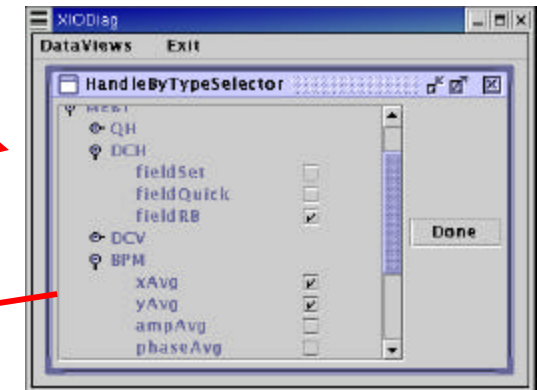
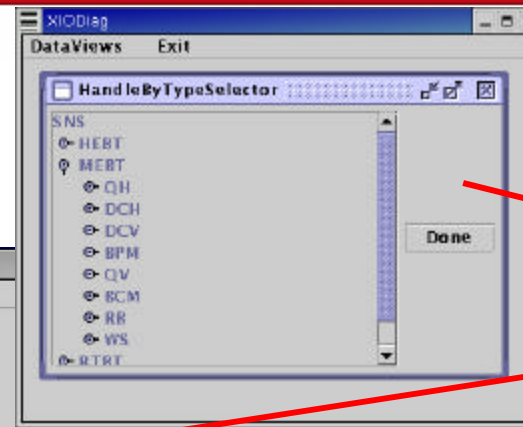
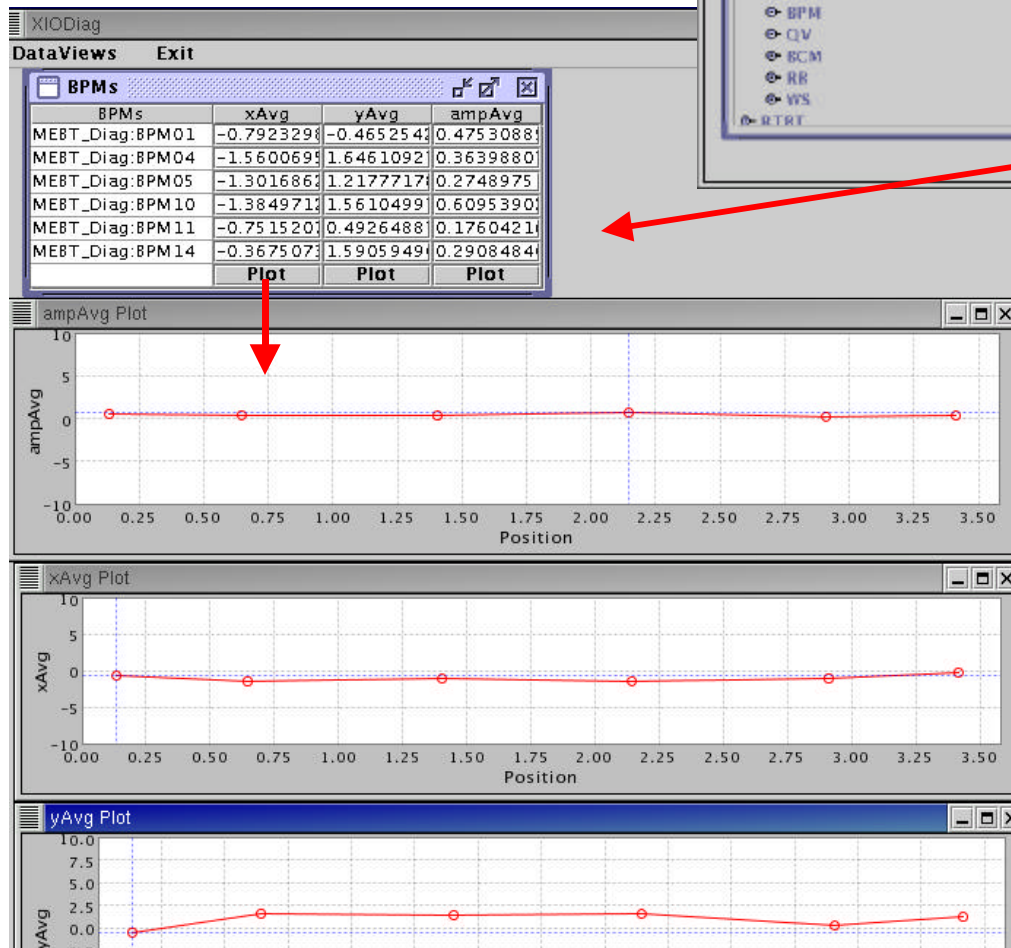


- Security
 - EPICS gateway server used at LBNL
 - Read/write access granted to ORNL Control room CPU IP address
- Additional communications - phone
 - We requested parameters to be tweaked, before we were given “set” privilege
 - Told us when to start / stop

Diagnostic App (N. Pattengale)



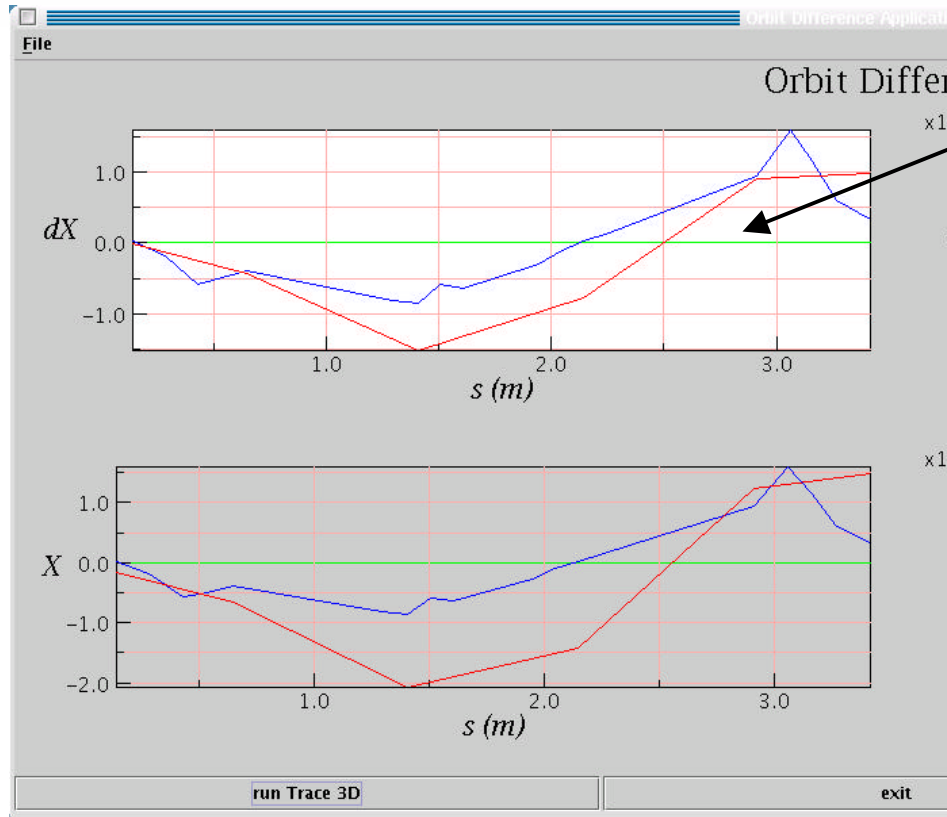
- General purpose: drill down accelerator hierarchy and tag items to be monitored



- Monitor accelerator values without having to know their names
- Works for any accelerator sequence

May 7-9, 2002

XAL Application: Model – Machine Comparison *(P. Chu)*



Diff. before/after correction

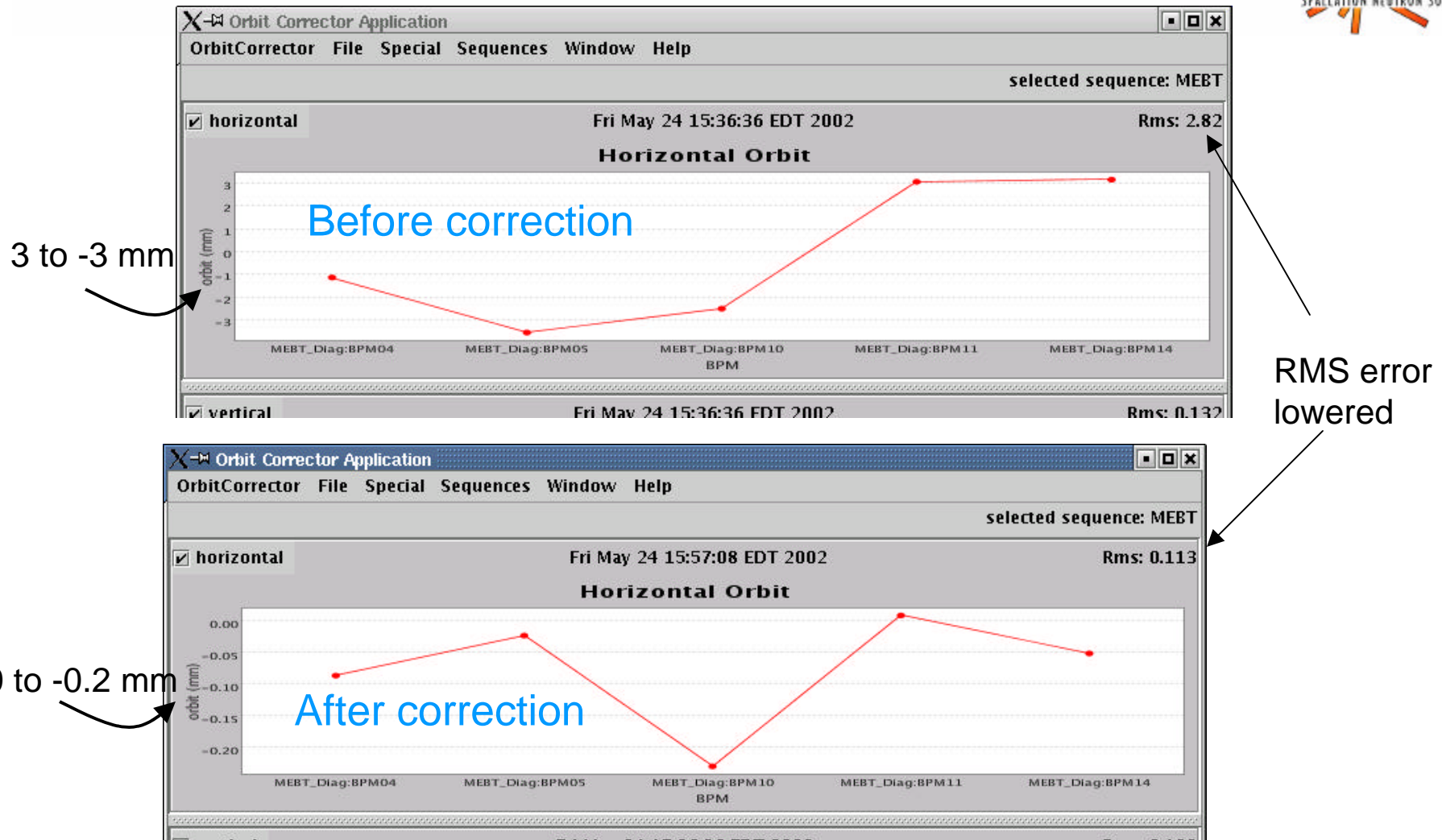
Using live “LBNL”
MEBT data

Red = BPM,
Blue = model

- Compare difference between two pulses
 - Observe effect of a magnet change
 - For both BPM signals and Model (Trace-3D)
- The beamline-device initialization is from the global database
- Also works with virtual accelerator

May 7–9, 2002

Orbit Correction application (app at ORNL, machine at LBNL)



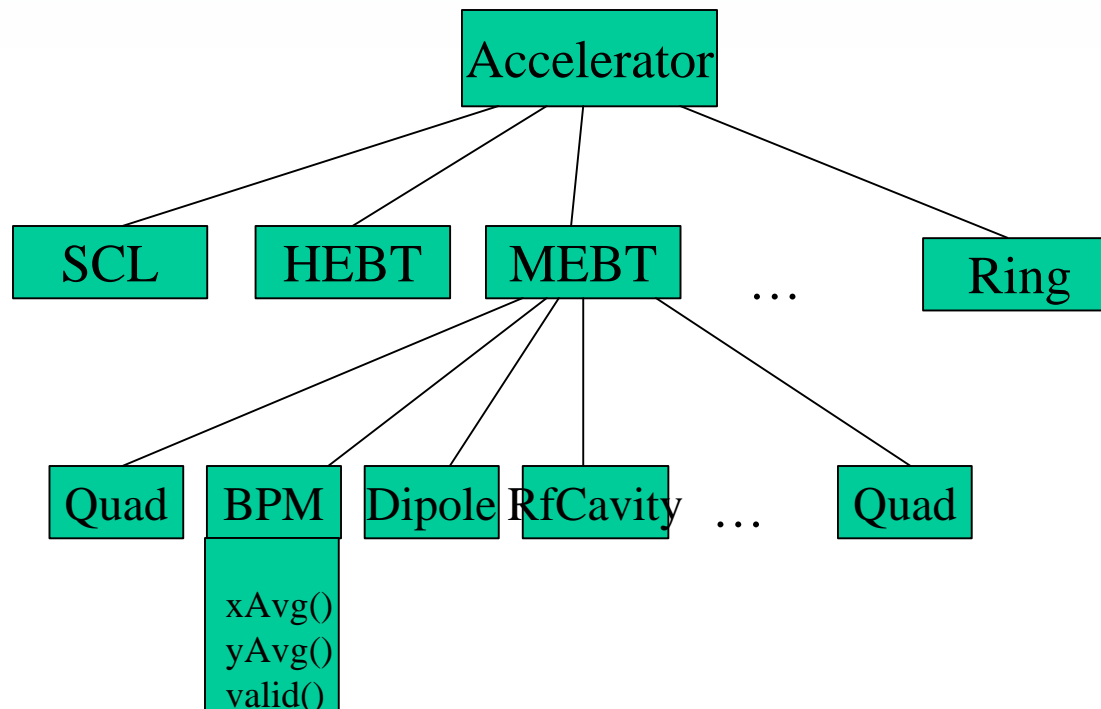
- 1st use of orbit correction application on real hardware

Thoughts on Remote Use of Applications (SNS)



- Requires interest on both ends + good communications
 - Needed a somewhat interested contact at LBNL
 - Took some “pestering” from ORNL side
 - Needed to know the “state” of diagnostics etc.
- Useful for commissioning
 - Difficult to accurately schedule when things will happen, avoid experts sitting around
 - Few hours of beam time was useful – able to identify some problems and see that some things worked
- News of its use can be greatly exaggerated
 - Bad: like to be honest
 - Good: indicates interest in demo-ing this technology
- Will partner labs remotely assist commissioning at ORNL?
 - Passive monitoring – undoubtedly
 - Active knob-turning – probably
 - Where there’s a will, there’s a way

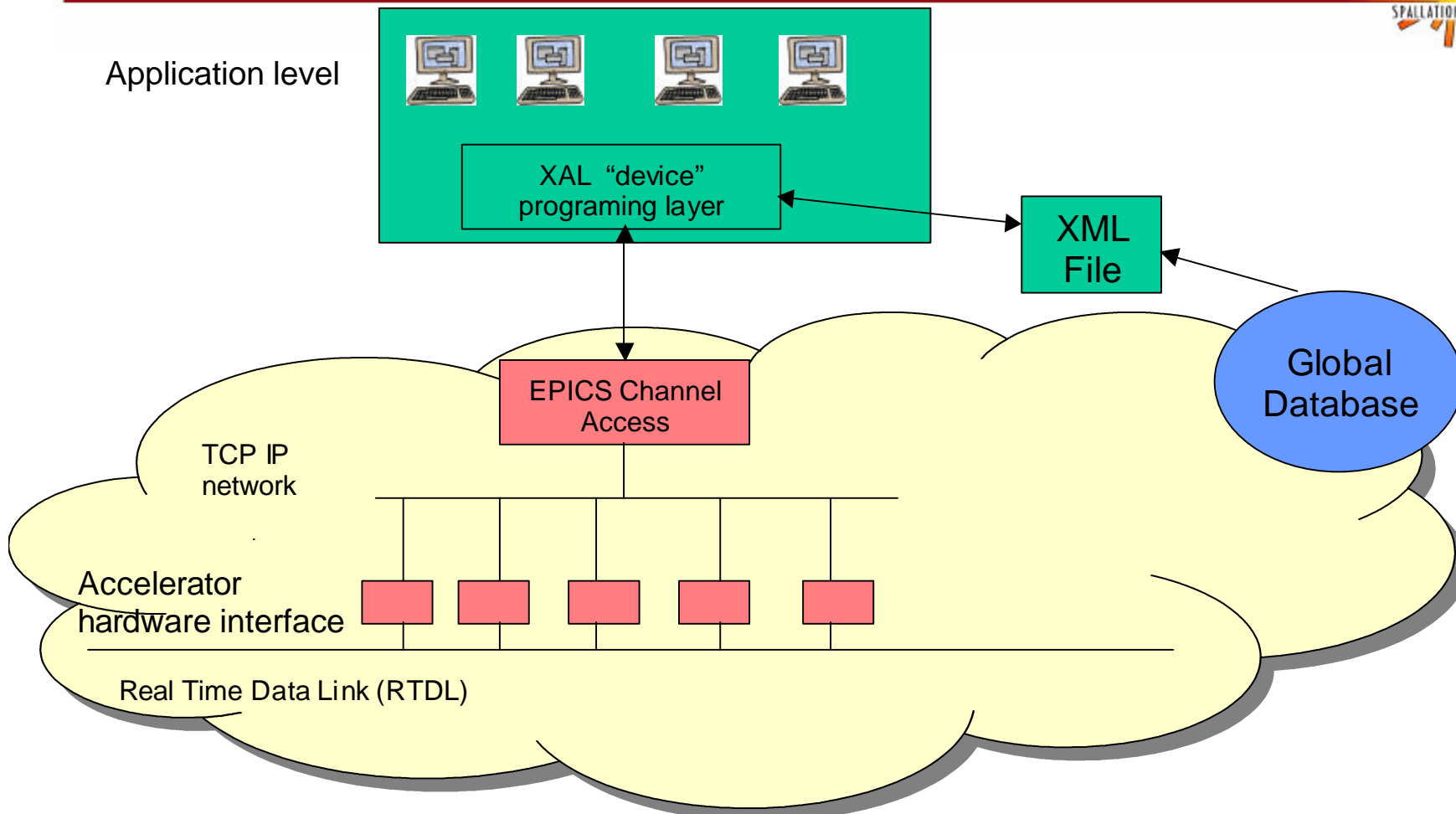
XAL – A Java based high level programming infrastructure for physics applications



- Java class structure that provides a hierarchical “device” view of the accelerator to the application programmers
- Setup from database, EPICS connections hidden
- Other similar frameworks
 - Based on UAL2 (<http://www.ual.bnl.gov/>)
 - Cosylab Abeans / databush (www.cosylab.com)

May 7–9, 2002

High Level Programming Infrastructure (XAL)



- XAL is a new programming infrastructure to provide a hierarchical device oriented view for application programmers
- Using Java - Interfaces exist to EPICS (java ca) and to database (JDBC + XML)

Advantages of Java (*for remote operations*)



- Simple
- Object Oriented
- Robust
 - Good memory management + exception handling
- *Multithreaded*
- *Architecture neutral*
- Interpreted + high performance
- *Distributed*
 - *Built in capabilities for TCP/IP and remote method invocation (client/server)*
- *Dynamic*
 - *Lots of run time information available – easy for applet deployment*
- Easier to attract good young programmers

Disadvantages of Java



- Less open software available for scientific applications
 - Graphics (contours, error bars, real-time, 3-D, ...)
 - Mathematical libraries less mature
- Java Virtual Machine (VM) somewhat of a hog, but:
 - With faster computers, matters less
 - Can run multiple applications from a single VM
- Have to teach old dogs new tricks